

PATENT ABSTRACTS OF JAPAN

(8)

(11)Publication number : 07-319471

(43)Date of publication of application : 08.12.1995

(51)Int.Cl.

G10H 7/00
G10H 1/18

(21)Application number : 06-324458

(71)Applicant : CASIO COMPUT CO LTD

(22)Date of filing : 27.12.1994

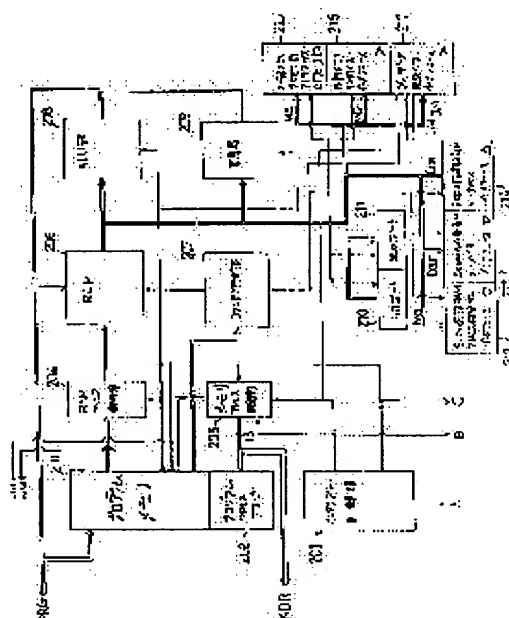
(72)Inventor : OKI KOICHIRO
USAMI RYUJI
SHIBA KOSUKE
OGURA KAZUO
HOSODA JUN

(54) MUSICAL SOUND WAVEFORM GENERATING DEVICE

(57)Abstract:

PURPOSE: To enable reducing the number of procession commands for every sound generating channel in the case of executing sound source processions making plural sound source systems coexist at every sound generating channel as software processions.

CONSTITUTION: When tones, etc., are changed and sound source systems to be executed at every sound generating channel are changed, a command analyzing part 207 and a memory address control part 205 load en bloc sound source processing programs in which sound source systems for 8 channels are defined from an external memory to a program memory 201. Then, since sound source programs in which sound source systems are automatically defined at every sound generating channel are executed, it is made unnecessary to perform a discrimination whether musical sounds are to be generated with which kinds of sound source systems at every sound generating channel.



LEGAL STATUS

[Date of request for examination] 27.12.1994

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number] 2678974

[Date of registration] 01.08.1997

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2000 Japan Patent Office

(P)

JP 07-319471A PARTIAL TRANSLATION

[0018]

The preferred embodiments of the present invention will now be described in detail hereinafter with reference to the accompanying drawings.

(Constitution of the preferred embodiment)

Referring to Fig. 1, there is shown an entire constitutional diagram of the preferred embodiment. Each of the components other than an external memory 116 is made of a single chip, in which two master and slave CPUs (central processing units) exchange information with each other, sharing a sound source process for generating musical sounds.

[0019]

In Fig. 1, first, the external memory 116 stores a musical sound control parameter such as a target value of an envelope value, a musical sound waveform in the PCM (pulse code modulation) system or a musical sound differential waveform in the DPCM (differential pulse code modulation) system, and programs for various sound source processes.

[0020]

On the other hand, the master CPU (hereinafter, simply referred to as MCPU) 101 and the slave CPU (hereinafter, simply referred to as SCPU) 102 access to above each data on the external memory 116 to share the sound source process. These CPUs share waveform data of the external memory 116 with each other and therefore in this condition a conflict

may occur when data is read from the external memory 116. Therefore, the MCPU 101 and the SCPU 102 output an address signal for an external memory access and external memory control data from an access address conflict preventive circuit 105 from output terminals 111 and 112 via an MCPU external memory access address latch section 103 and an SCPU external memory access address latch section 104, respectively, thereby enabling a conflict between an address from the MCPU 101 and an address from the SCPU 102.

[0021]

Data read from the external memory 116 on the basis of the above addressing is inputted from the external memory data in-terminal 115 to an external memory selector section 106. The external memory selector section 106 separates the above read data into data inputted to the MCPU 101 through a data bus MD and data inputted to the SCPU 102 through a data bus SD on the basis of a control signal from the access address conflict preventive circuit 105 and then inputs them to the MCPU 101 and the SCPU 102, respectively. This prevents a conflict of data.

[0022]

Thereafter, a sound source process is performed for each data with software in the MCPU 101 and the SCPU 102, all data of the sound generation channels is accumulated, and a left analog output of a left channel and a right analog output of a right channel are outputted as musical sound signals from a left output terminal 113 of a left digital-to-

analog converter section 107 and a right output terminal 114 of a right digital-to-analog converter section 108, respectively.

[0023]

Next, referring to Fig. 2, there is shown a block diagram of an internal constitution of the MCPU 101. In Fig. 2, a program memory 201 comprises a ROM portion for storing a program corresponding to a main flow (Fig. 9) described later and a RAM portion for loading a sound source processing program described later from the external memory 116 and storing it and sequentially outputs program words (instructions) of specified addresses from a memory address control section 205 via a program address decoder 202. Specifically, each program word has a word length of 28 bits, for example, and is applied a next address system in which a part of the program word is inputted to the memory address control section 205 as a low order part of the address (intra-page address) to be read next.

[0024]

This embodiment is characterized by that, in case of a change of a sound source system executed for each sound generation channel caused by a change of a tone or the like, corresponding sound source processing programs are collectively loaded from the external memory 116 in Fig. 1 to the program memory 201.

[0025]

A command analysis section 207 analyzes an operation

code of an instruction outputted from the control RAM 201 and transmits control signals to respective portions of the circuit to execute a specified operation.

[0026]

When an operand of the instruction from the control RAM 201 specifies a register, a RAM address control section 204 specifies an address of the corresponding register in a RAM 206. The RAM 206 stores various musical sound control data described later with reference to Fig. 15 and Fig. 16 for 8 sound generation channels and various buffers or the like described later with reference to Fig. 17 for use in a sound source process described later.

[0027]

If an instruction from the control RAM 201 is an arithmetic instruction, an ALU section 208 executes an addition, a subtraction, and a logical operation and a multiplier 209 executes a multiplication on the basis of designations from the command analysis section 207.

[0028]

An interrupt control section 203 supplies a reset cancel signal A to the SCPU 102 in Fig. 1 and interrupt signals to the memory address control section 205 and to the digital-to-analog converter sections 107 and 108 in Fig. 1 at fixed time intervals on the basis of an internal component, specifically, a hard timer which is not shown.

[0029]

In addition to the above constitution, the MCPU 101 in

Fig. 2 is provided with various interfaces of buses described below. In other words, it has an interface 215 of an address bus MA for specifying an address of the external memory 116 to access it, an interface 216 of a data bus MD for exchanging accessed data with the MCPU 101 via the external memory selector section 106, an interface 212 of a bus Ma for specifying an address of the RAM inside the SCPU 102 to execute the exchange of data with the SCPU 102, an interface 213 of a data bus Dout for the MCPU 101 to write data into the SCPU 102, an interface 214 of a bus Din for the MCPU 101 to read data from the SCPU 102, an interface 217 of a digital-to-analog data transfer bus for transferring final output waveforms to the left digital-to-analog converter section 107 and the right digital-to-analog converter section 108, and input-output ports 210 and 211 for exchanging data with an external switch section, a keyboard section (see Fig. 7 and Fig. 8), or the like.

[0030]

Next, an internal constitution of the SCPU 102 is shown in Fig. 3. The SCPU 102 only performs a sound source process in response to a process start signal from the MCPU 101 and therefore it has no interrupt control section corresponding to 223 in Fig. 2, no input-output port for exchanging data with an external circuit corresponding to 210 and 211 in Fig. 2, and no interface for outputting musical sound signals to the left digital-to-analog converter section 107 and the right digital-to-analog converter section 108 corresponding

to 217 in Fig. 2. Other circuits 301, 302, and 304 to 309 have the same functions as those of the circuits 201, 202, and 204 to 209 in Fig. 2. Interfaces 303, and 310 to 313 are arranged so as to be opposed to the circuits 212 to 216 in Fig. 2. An address of the RAM inside the SCPU 102 specified via the bus Ma by the MCPU 101 is inputted to the RAM address control section 304 and the corresponding address is specified for the RAM 306. This causes accumulated waveform data of the maximum 8 sound generation channels, which has been generated in the SCPU 102 and retained in the RAM 306, to be outputted to the MCPU 101 via the data bus Din. It will be described later.

[0031]

At a change of a tone or the like, the corresponding sound source processing programs are collectively loaded from the external memory 116 shown in Fig. 1 via the MCPU 101 in the same manner as for the MCPU 101 as set forth in the above. This transfer operation is executed by the command analysis section 207 and the memory address control section 205 in the MCPU 101 in Fig. 2 controlling the program RAM 310 and the RAM address control section 305 via a gate circuit 314.

[0037]

In this embodiment, basically the MCPU 101 leads the operations and repeats a series of processes from S902 to S910 as shown in a main flowchart in Fig. 9. An actual sound

source process is performed as an interrupt process. Specifically, an interrupt is made to the MCPU 101 and the SCPU 102 at fixed time intervals and the respective CPUs perform a sound source process for generating sounds of the maximum 8 channels.

[0038]

While it is possible to allocate different sound source systems such as PCM, DPCM, FM or TM, and the like for the respective sound generation channels as described later, the sound source processing programs of the 8 channels corresponding to the allocations are collectively loaded from the external memory 116 (Fig. 1) to the program memory 201 (Fig. 2) or the program RAM 301 at the tone setting, which characterizes this embodiment. Accordingly, there is no need for a determination process of which sound source system should be adopted to generate a musical sound for each sound generation channel. This transfer operation will be described later.

[0039]

When the sound source process with the interrupt processes is completed, the musical sound waveforms are added for the maximum 16 channels totaled by each of the maximum 8 channel of the respective CPUs and then outputted from the left digital-to-analog converter section 107 and the right digital-to-analog converter section 108. Thereafter, the control returns from the interrupt status to the main flow. The above interrupts are made periodically on the basis of

the hard timer in the interrupt control section 203 in Fig. 2. This period is equal to a sampling period at an output of a musical sound.

[0096]

In the FM modulation system, normally hardware or software modules having the same function called operator as designated by OP1 and OP2 in Fig. 24 are used and they are connected with each other conforming to connection rules as shown in the diagram, by which musical sounds are generated. In this embodiment, the FM modulation system is realized by software processes.

[0097]

Referring to Fig. 23, there is shown an operation flowchart for executing the sound source process in the FM modulation system with two operators. An algorithm of the process is shown in Fig. 24. Variables in the flowchart are data of the FM format on a table 1 in Fig. 16, which should be stored in one of the sound generation channel regions in Fig. 14 on the RAM 206 or 306 of the MCPU 101 or the SCPU 102.

[0098]

A process of an operator 2 (OP2), which is a modulator, is performed first. A pitch process is performed only for an integer address A2 due to no interpolation as performed in the PCM system. In other words, it is assumed that the external memory 116 (Fig. 1) stores waveform data for a

modulation at sufficiently fine stepping intervals.

[0099]

First, pitch data P2 is added to the current address A2 (S2401). Next, a feedback output F02 is added as a modulation input to the address A2, by which a new address AM2 is obtained (S2402). The feedback output F02 is obtained by an execution of a process in S2405 described later at the previous interrupt timing.

[0100]

Furthermore, a value of a sine wave corresponding to the address AM2 (phase) is calculated. Actually sine-wave data is stored in the external memory 116 (Fig. 1) and therefore the value is obtained by referring to the table to find the sine-wave data with the address AM2 (S2403).

[0101]

Subsequently, an output O2 is obtained by multiplying the sine-wave data by an envelope value E2 (S2404). Thereafter, the output O2 is multiplied by a feedback level FL2, by which a feedback output F02 is obtained (S2405). This output F02 is inputted to an operator 2 (OP2) at the next interrupt timing in this embodiment.

[0102]

In addition, O2 is multiplied by a modulation level ML2, by which a modulation output MO2 is obtained (S2406). This modulation output MO2 is a modulation input to an operator 1 (OP1).

[0103]

Next, the control shifts to a process of the operator 1 (OP1). This process is almost the same as one for the operator 2 in the above except that there is no modulation input of the feedback output. First, pitch data P1 is added to the current address A1 of the operator 1 (S2407) and then the above modulation output MO2 is added to this value, by which a new address AM1 is obtained (S2408).

[0104]

Next, a value of a sine wave corresponding to this address AM1 (phase) is read from the external memory 116 (Fig. 1) (S2409) and it is multiplied by the envelope value E1, by which a musical sound waveform output O1 is obtained (S2410).

[0105]

Then, it is added to a buffer B (see Fig. 17) in the RAM 206 (Fig. 2) or 306 (Fig. 3) (S2411) and the FM modulation process of a sound generation channel is completed.

[Fig. 1]

Entire constitutional diagram of embodiment according to this invention

101: Master central processing unit (MCPU)

102: Slave central processing unit (SCPU)

103: MCPU external memory access address latch section

104: SCPU external memory access address latch section

105: Access address conflict preventive circuit

106: External memory selector section

107: Left D-A converter section
108: Right D-A converter section
116: 1. Sound source control data
2. Waveform data
3. Storage element (ROM) storing various sound source processing programs
入力ポート : Input port
出力ポート : Output port
外部メモリアクセス用アドレス信号 : External memory access address signal
外部メモリ制御データ : External memory control data
外部メモリデータイン : External memory data IN
レフト・アナログ出力 : Left analog output
ライト・アナログ出力 : Right analog output

[Fig. 2]

Master CPU internal constitutional diagram

201: Program memory
202: Program address decoder
203: Interrupt control section
204: RAM address control section
205: Memory address control section
207: Command analysis section
208: ALU section
209: Multiplier
210: Input port
211: Output port

212: Bus interface for specifying address of RAM in SCPU
213: Data bus interface for writing to SCPU
214: Data bus interface for reading from SCPU
215: Address bus interface for external memory access
216: External memory data bus interface
217: D-A data transfer bus interface

[Fig. 3]

Slave CPU internal constitutional diagram

301: Program RAM
データイン: Data IN
データアウト: Data OUT
302: ROM address decoder
303: Bus interface for specifying address of RAM in MCPU or
SCPU
304: RAM address control section
305: ROM address control section
307: Command analysis section
308: ALU section
309: Multiplier
310: Data bus interface for writing from MCPU
311: Data bus interface for reading to MCPU
312: Address bus interface for external memory access
313: External memory data bus interface
314: Gate circuit

[Fig. 9]

Main operation flowchart

MCPU start

S901: Power-on initialization

S902: Functional key fetching

S903: Functional key process

S904: Keyboard key fetching

S905: Keyboard key process

S906: Demonstration performance process

S907: Rhythm process

S908: Timer process

S909: Sound generation process

S910: Single flow cycle preparation process

[Fig. 14]

Storage area according to sound generation channel on RAM

[Fig. 24]

Algorithm of sound source process in FM system

[Fig. 16]

Constitutional diagram of data format according to sound source system on RAM (2)

FM

A1: Current address (OP1)

A2: Current address (OP2)

P1: Pitch data (OP1)

P2: Pitch data (OP2)
E1: Envelope (OP1)
E2: Envelope (OP2)
ML2: Modulation level (OP2)
MO2: Modulation output (OP2)
FL2: Feedback level (OP2)
FO2: Feedback output (OP2)
O1: OP1 output
O2: OP2 output

[Fig. 23]

Operation flowchart of sound source process in FM system

FM system process

S2301: Pitch data added (式挿入：以下同様)
S2302: Feedback output added
S2303: Sine-wave conversion
S2304: Envelope
S2305: Feedback output
S2306: Modulation output
S2307: Pitch data added
S2308: Modulation input
S2309: Sine-wave conversion
S2310: Envelope
S2311: Added to buffer

(8)

(19)日本国特許庁 (J P)

(12) 公 開 特 許 公 報 (A)

(11)特許出願公開番号

特開平7-319471

(43)公開日 平成7年(1995)12月8日

(51)Int.Cl. ⁶	識別記号	庁内整理番号	F I	技術表示箇所
G 1 0 H 7/00	5 1 2			
1/18	1 0 1			

審査請求 有 請求項の数 2 O L (全 24 頁)

(21)出願番号 特願平6-324458
実願平2-402562の変更
(22)出願日 平成2年(1990)12月28日

(71)出願人 000001443
カシオ計算機株式会社
東京都新宿区西新宿2丁目6番1号
(72)発明者 太期 広一郎
東京都西多摩郡羽村町栄町3丁目2番1号
カシオ計算機株式会社羽村技術センター
内
(72)発明者 宇佐美 隆二
東京都西多摩郡羽村町栄町3丁目2番1号
カシオ計算機株式会社羽村技術センター
内
(74)代理人 弁理士 大曾 義之

最終頁に続く

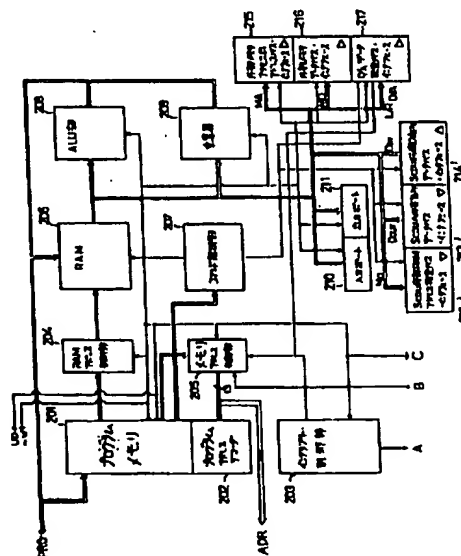
(54)【発明の名称】 楽音波形発生装置

(57)【要約】

【目的】 発音チャンネル毎に複数の音源方式を混在させた音源処理をソフトウェア処理として実行する場合に、発音チャンネル毎の処理命令数の削減を可能とすることを目的とする。

【構成】 音色等が変更されて発音チャンネル毎に実行される音源方式が変更された場合に、コマンド解析部207及びメモリアドレス制御部205は、外部メモリからプログラムメモリ201に、8チャンネル分の音源方式が定まった音源処理プログラムを一括してロードする。そして、各発音チャンネル毎に、自動的に音源方式が定まった音源処理プログラムが実行され、各発音チャンネル毎にどの音源方式で楽音を生成するかという判別処理を行うことが不要となる。

図2- CPUの内部構成図



【特許請求の範囲】

【請求項1】 所定時間間隔で、複数の各発音チャンネル毎に、複数の音源方式のうち任意の音源方式の音源処理プログラムを実行して楽音信号を生成するプログラム実行手段を含む楽音波形発生装置において、

前記全発音チャンネル分連続して実行される音源処理プログラムを、前記全発音チャンネルの音源方式の組合せ毎に複数組保存する保存用プログラム記憶手段と、

前記所定時間間隔毎に、前記プログラム実行手段により、前記全発音チャンネル分連続して実行される音源処理プログラムを記憶する実行用プログラム記憶手段と、

前記保存用プログラム記憶手段から前記全発音チャンネル分連続して実行される音源処理プログラムの任意の組を読み出して前記実行用プログラム記憶手段に転送する音源処理プログラム転送制御手段と、

を有することを特徴とする楽音波形発生装置。

【請求項2】 所定時間間隔で、複数の各発音チャンネル毎に、複数の音源方式のうち任意の音源方式の音源処理プログラムを実行して楽音信号を生成するプログラム実行手段を含む複数のプロセッサで構成される楽音波形発生装置において、

前記各プロセッサ毎に全発音チャンネル分連続して実行される音源処理プログラムを、前記全発音チャンネルの音源方式の組合せ毎に複数組保存する保存用プログラム記憶手段と、

前記各プロセッサ毎に設けられ、前記所定時間間隔毎に、前記各プロセッサのプログラム実行手段により、前記各プロセッサの全発音チャンネル分連続して実行される音源処理プログラムを記憶する実行用プログラム記憶手段と、

前記保存用プログラム記憶手段から前記全発音チャンネル分連続して実行される音源処理プログラムの任意の組を読み出して前記各プロセッサの実行用プログラム記憶手段に転送する音源処理プログラム転送制御手段と、

を有することを特徴とする楽音波形発生装置。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 本発明は、楽音波形発生装置における音源処理方式に関し、更に詳しくは、発音チャンネル毎に複数の音源方式を混在させた音源処理をソフトウェア処理として実行する楽音波形発生装置に関する。

【0002】

【従来の技術】 デジタル信号処理技術とLSI処理技術の発達により性能の良い様々な電子楽器が実現されている。

【0003】 電子楽器の楽音波形発生装置は、大量かつ高速のデジタル演算が必要なため、従来は、必要とする音源方式に基づく楽音発生アルゴリズムと等価なアーキテクチャをハードウェアで実現した専用の音源回路によって構成されていた。そのため、ハードウェア規模が

大きくなってしまい、また、音源方式の自由な変更が困難であり、更に、このような楽音波形発生装置を電子楽器として実現するような場合に、演奏情報に対する処理と音源に対する処理の両方を同期させて行うための制御が複雑となり、その開発において多大なコストアップを招いていた。

【0004】 その一方、近年においては、汎用のデータ処理を行うための高性能なマイクロプロセッサが多く実現されており、このようなマイクロプロセッサを使用して音源処理をソフト的に行う楽音波形発生装置を実現させることも考えられる。

【0005】 このような方式では、例えばプロセッサが、通常、演奏情報処理プログラムを実行して演奏操作に基づいて鍵盤等から発生する演奏情報を処理し、所定の時間間隔で割り込みがかかることによって、音源処理プログラムに処理を移して楽音生成を行う。その後、音源処理プログラムの実行が終了すると、割り込みが解除されて演奏情報処理プログラムの実行が再開される。

【0006】 上述のようなソフトウェア音源処理においては、複数の楽音を同時に発音可能とすべく、一般に、所定時間間隔の割り込み等の処理周期中に複数の発音チャンネルに対して時分割処理を行うようにしている。そして、各発音チャンネル毎に異なる音源方式の音源処理プログラムを実行させることにより、ある音色のときはPCM方式で、別の音色のときは変調方式で、或いは、ある音色のときに偶数チャンネルはPCM方式で、奇数チャンネルは変調方式でというように、音色毎又は発音チャンネル毎に異なる音源方式によって楽音を生成することも可能である。

【0007】 このような場合、一般的に、発音チャンネル毎に音源方式の番号等をメモリなどに記憶させておき、各発音チャンネルの処理タイミングにさしかかる毎に、その発音チャンネルに対応して記憶されている音源方式番号等を識別して、対応する音源方式の音源処理プログラムを実行するような制御が行われる。

【0008】 ここで、楽器の性能を向上させるために、同時に発音可能な発音チャンネル数を増加させたいというような要求があった場合、上記処理周期中にできるだけ多くのプログラム命令を実行できるようにする必要がある。

【0009】

【発明が解決しようとする課題】 従って、各発音チャンネルの処理においては、分岐命令等の演算サイクルの長い命令はできる限り削減する必要が生じる。しかし、発音チャンネル毎に異なる音源方式による楽音生成を可能としたい場合に、上述の従来例のように、各発音チャンネルの処理タイミングにさしかかる毎に音源方式を識別するような分岐命令を実行していたのでは、1処理期間中に全ての発音チャンネルに対する処理を終了できなくなる可能性はある。

【0010】そして、それを防止するために、楽音を発生させるサンプリング周波数を減少させ音源処理の時間間隔を大きくするなどの措置を講じなければならなくなり、その結果、楽音を発生させるサンプリング周波数が減少して生成される楽音の最高周波数が低くなり音質が劣化してしまうなどの問題点を生じていた。

【0011】本発明の課題は、発音チャンネル毎に複数の音源方式を混在させた音源処理をソフトウェア処理として実行する場合に、発音チャンネル毎の処理命令数の削減を可能とすることにある。

【0012】

【課題を解決するための手段】本発明は、所定時間間隔で、複数の各発音チャンネル毎、例えば8チャンネルのうちのそれぞれのチャンネル毎に、複数の音源方式、例えばPCM方式、DPCM方式、FM方式、TM方式等のうち、任意の音源方式の音源処理プログラムを実行して楽音信号を生成するプログラム実行手段を含む楽音波形発生装置を前提とする。この場合、複数のプロセッサ構成として、より多くの発音チャンネルを処理可能としてもよい。

【0013】そして、まず、全発音チャンネル分連続して実行される音源処理プログラムを、全発音チャンネルの音源方式の組合せ毎に複数組保存するROM等の保存用音源処理プログラム記憶手段を有する。同手段には、例えば、8チャンネル全てがPCM方式で楽音生成を連続して行う音源処理プログラム、各発音チャンネル毎にPCM方式、TM方式等が混在した形態で楽音生成を連続して行う音源処理プログラム等、複数組が記憶される。

【0014】次に、前述の所定時間間隔毎に、プログラム実行手段により、全発音チャンネル分連続して実行される音源処理プログラムを記憶するRAM等の実行用プログラム記憶手段を有する。なお、前述のように複数プロセッサ構成とした場合には、各プロセッサ毎に設けられる。

【0015】そして、保存用プログラム記憶手段から全発音チャンネル分連続して実行される音源処理プログラムの任意の組を読み出して実行用プログラム記憶手段に転送する音源処理プログラム転送制御手段を有する。なお、前述のように複数プロセッサ構成とした場合には、同制御手段は、各プロセッサの実行用プログラム記憶手段に転送を行う。

【0016】

【作用】演奏者が音色設定等を行った場合又は装置の電源をオンしたような場合に、音源処理プログラム転送制御手段が、保存用プログラム記憶手段から全発音チャンネル分連続して実行される音源処理プログラムの任意の組を読み出して実行用プログラム記憶手段に転送する。そして、プログラム実行手段は、所定時間間隔毎に各発音チャンネルの楽音生成を行う場合、実行用プログラム記憶手段に記憶された音源処理プログラムを全発音チャンネル

分一括して実行する。

【0017】これにより、各発音チャンネル毎にどの音源方式で楽音を生成するかという判別処理を行うことが不要となり、分岐命令の数を削減することができるため、同時に発音可能な発音チャンネル数を増加させることができる等、楽器の性能を向上させることが可能となる。

【0018】

【実施例】以下、図面を参照しながら本発明の実施例につき説明する。

＜本実施例の構成＞図1は、本実施例の全体構成図であり、外部メモリ116以外は1チップで構成され、その中のマスターとスレーブの2つのCPU（中央演算制御装置）が互いに情報を交換しつつ、楽音作成のための音源処理を分担して行う。

【0019】図1において、まず、外部メモリ116には、エンベロープ値の目標値等の楽音制御パラメータと、PCM（パルス符号変調）方式における楽音波形又はDPCM（差分パルス符号変調）方式における楽音差分波形、ならびに各種音源処理のためのプログラム等が記憶されている。

【0020】一方、マスタCPU（以下、MCPUと略称する）101とスレーブCPU（以下、SCPUと略称する）102は、外部メモリ116上の上記各データをアクセスして、分担して音源処理を行う。これらのCPUは、ともに外部メモリ116の波形データ等を共用するので、このままでは、外部メモリ116からデータを読み込むときに、競合が発生する恐れがある。そのため、MCPU 101とSCPU 102のそれぞれは、MCPU 外部メモリアクセス用アドレスラッチ部103及びSCPU 外部メモリアクセス用アドレスラッチ部104を介して、アクセス用アドレス競合回避回路105から外部メモリアクセス用のアドレス信号と外部メモリ制御データを出力端子111、112から出力することにより、MCPU 101からのアドレスとSCPU 102からのアドレスの競合を回避することができる。

【0021】上記アドレス指定に基づいて外部メモリ116から読み出されたデータは、外部メモリデータイン端子115から外部メモリセクタ部106に入力される。外部メモリセクタ部106は、アクセス用アドレス競合回避回路105からの制御信号に基づき、上記読み出されたデータを、データバスMDを通してMCPU 101に入力されるデータと、データバスSDを通してSCPU 102に入力されるデータに分離し、それぞれMCPU 101とSCPU 102に入力させる。これにより、データの競合も回避することができる。

【0022】その後、それぞれのデータに対し、MCPU 101及びSCPU 102で、ソフトウェアによって音源処理が施された後、発音チャンネル分全部が累算され、Left D/A変換器部107のレフト出力端子113及びRight D/A変換器部108のライト出力端子114か

ら、それぞれ楽音信号として、左チャンネルのレフト・アナログ出力と右チャンネルのライト・アナログ出力が出力される。

【0023】次に、図2はMCPU 101の内部構成を示すブロック図である。同図において、プログラムメモリ201は、後述するメインフロー（図9）に対応するプログラムを記憶するROM部分と、後述する音源処理プログラムを図1の外部メモリ116からロードして記憶するRAM部分とからなり、メモリアドレス制御部205からプログラムアドレスデコーダ202を介して指定されたアドレスのプログラム語（命令）を順次出力する。具体的には、各プログラム語の語長は例えば28ビットであり、プログラム語の一部が次に読み出されるべきアドレスの下位部（ページ内アドレス）としてメモリアドレス制御部205に入力されるネクストアドレス方式となっている。

【0024】なお、音色等が変更されて発音チャンネル毎に実行される音源方式が変更された場合に、対応する音源処理プログラムを一括して図1の外部メモリ116からプログラムメモリ201にロードするようにした点が、本実施例の特徴とする点である。

【0025】コマンド解析部207は、制御用RAM201から出力される命令のオペコードを解析し、指定されたオペレーションを実行するために、回路の各部に制御信号を送る。

【0026】RAMアドレス制御部204は、制御用RAM201からの命令のオペランドがレジスタを指定している場合に、RAM206内の対応するレジスタのアドレスを指定する。RAM206には、図15、図16等として後述する各種楽音制御データが8発音チャンネル分記憶されるほか、図17等として後述する各種バッファ等が記憶され、後述する音源処理に使用される。

【0027】ALU部208及び乗算器209は、制御用RAM201からの命令が演算命令の場合に、コマンド解析部207からの指示に基づいて、前者は加減算と論理演算、後者は乗算を実行する。

【0028】インタラプト制御部203は、内部の特には図示しないハードタイマに基づいて、一定時間毎に、図1のSCPU 102にリセット解除信号A、メモリアドレス制御部205及び図1の各D/A変換器部107、108にインタラプト信号を供給する。

【0029】上述の構成のほかに、図2のMCPU 101には、次の各種のバスに関するインタフェースが設けられている。すなわち、外部メモリ116をアクセスすべく、そのメモリのアドレスを指定するためのアドレスバスMAのインタフェース215、アクセスされたデータを外部メモリセクタ部106を介してMCPU 101との間で授受するためのデータバスMDのインタフェース216、SCPU 102とのデータの授受を実行すべくSCPU 102内部のRAMのアドレスを指定するバスMa

のインタフェース212、MCPU 101がSCPU 102へデータを書き込むためのデータバスDoutのインタフェース213、MCPU 101がSCPU 102からデータを読み込むためのデータバスDinのインタフェース214、Left D/A変換器部107、Right D/A変換器部108に最終出力波形を転送するためのD/Aデータ転送バスのインタフェース217、及び外部のスイッチ部又は鍵盤部（図7、図8参照）等との間でデータの授受を行う入出力ポート210、211がある。

【0030】次に、SCPU 102の内部構成を図3に示す。SCPU 102は、MCPU 101からの処理開始信号を受けて音源処理を行うのみなので、図2の223に対応するインタラプト制御部、図2の210及び211に対応する外部回路とのデータの授受を行う入出力ポート及び図2の217に対応するLeft D/A変換器部107とRight D/A変換器部108に楽音信号を出力するためのインタフェースはない。それ以外の301、302、304～309の各回路は、図2の201、202、204～209の各回路と同じ機能を有する。また、各インタフェース303、310～313は、図2の212～216のそれぞれに対向して設けられる。なお、MCPU 101からバスMaを介して指定されたSCPU 102内部RAMアドレスは、RAMアドレス制御部304に入力し、対応するアドレスがRAM306に対して指定される。これによって、例えばSCPU 102で生成されRAM306内部に保持されている最大8発音チャンネル分の累算波形データが、データバスDinを介してMCPU 101に出力される。これについては後述する。

【0031】また、音色変更時等においては、前述したMCPU 101の場合と同様、対応する音源処理プログラムが一括して図1の外部メモリ116からMCPU 101を介しロードされる。この転送動作は、図2のMCPU 101内のコマンド解析部207及びメモリアドレス制御部205等が、ゲート回路314を介して、プログラムRAM301及びRAMアドレス制御部305を制御することにより実行される。

【0032】以上に示される構成のほか、本実施例では、MCPU 101の入力ポート210に、図7及び図8に示されるような機能キー701及び鍵盤キー702等が接続される。これらの部分が、実質的な楽器操作部を構成する。

【0033】次に、図5は、図1のLeftとRightのD/A変換器部107、108（両変換部の内容は同じ）の内部構成を示すもので、データバスを介して、音源処理で作成された楽音の1サンプルデータがラッチ401に入力される。そして、ラッチ401のクロック入力にMCPU 101のコマンド解析部207（図2）から音源処理終了信号が入力されると、データバス上の1サンプル分の楽音データがラッチ401にラッチされる。

【0034】ここで、前述の音源処理に要する時間は、音源処理用のソフトウェアにより変化する。そのため、音源方式が異なる場合は、各音源処理が終了し、ラッチ401に楽音データがラッチされるタイミングは一定でない。そのため、図4のように、ラッチ401の出力をそのままD/A変換器402に入力させることはできない。

【0035】そこで、本実施例では図5の如く、ラッチ401の出力を更にラッチ501でラッチし、インタラプト制御部203から出力されるサンプリングクロック間隔に等しい図6に示されるインタラプト信号により、楽音信号をラッチ501にラッチさせ、図6のように一定間隔でD/A変換器402に出力させるようにしている。

【0036】このようにラッチを2つ用いて、音源方式による処理時間の変化を吸収したので、楽音データをD/A変換器へ出力させるための複雑なタイミング制御プログラムが不用になった。

＜本実施例の全体動作＞次に、本実施例の全体動作を説明する。

【0037】本実施例は、基本的にはMCPU 101が中心となって動作し、図9のメインフローチャートに示すように、S902～S910の一連の処理を繰り返し行っている。そして実際の音源処理は割り込み（インタラプト）処理で行っている。具体的には、ある一定時間毎に、MCPU 101とSCPU 102に割り込みが掛かり、それに基づいてそれぞれのCPUが最大8チャンネルずつの音を作る音源処理を行う。

【0038】ここで、後述するように、各発音チャンネル毎に、PCM、DPCM、FM又はTM等の異なる音源方式を割り当てることが可能であるが、その割り当てに対応する8チャンネル分の音源処理プログラムは、音色設定時等に一括して外部メモリ116（図1）からプログラムメモリ201（図2）又はプログラムRAM301にロードされることが、本実施例の特徴となっている。これにより、各発音チャンネル毎にどの音源方式で楽音を生成するかという判別処理を行うことが不要となっている。なお、この転送動作については、後述する。

【0039】インタラプト処理による音源処理が終わると、それぞれのCPUの最大8チャンネル、計最大16チャンネル分の楽音波形が加算され、Left D/A変換器部107、Right D/A変換器部108から出力される。その後、割り込み状態からメインフローに戻る。なお、上述の割り込みは、図2のインタラプト制御部203内のハードタイマに基づき周期的に行われる。この周期は楽音出力時のサンプリング周期に等しい。

【0040】以上が、本実施例の概略の全体動作である。次に、本実施例の全体動作の詳細について、図9～図11を用いて説明する。図9のメインフローチャートのS902～S910の処理が繰り返し実行されている間

に、インタラプト制御部203から割り込みが掛かると、図10のMCPU インタラプト処理と図11のSCPU インタラプト処理の2つの処理が同時に起動する。そして、図10と図11の「音源処理」は、後述する図13に示される。

【0041】図9のメインフローチャートは、インタラプト制御部203から割り込みが掛からない状態においてMCPU 101において実行される、音源処理以外の処理の流れを示している。

【0042】まず、電源がONされると、MCPU 101のRAM206の内容等の初期設定が行われる（S901）。次に、MCPU 101の外部に接続される機能キー701（図7参照）、例えば音色スイッチ等が走査され（S902）、各スイッチの状態が入力ポート210からRAM206内のキーバッファエリアに取り込まれる。その走査の結果、状態の変化した機能キーが識別され、対応する機能の処理がなされる（S903）。例えば、楽音番号のセット、エンベロープ番号のセット、また、付加機能にリズム演奏がついていれば、リズム番号のセット等が行われる。

【0043】その後、押鍵されている鍵盤キーの状態が上述の機能キーの場合と同様にに取り込まれ（S904）、変化した鍵（押鍵又は離鍵された鍵）が識別されることによりキーアサイン処理が行われる（S905）。この処理は、押鍵された鍵に基づいて発音されるべき楽音のデータを発音チャンネルに割り当てたり、逆に、離鍵された鍵に対応する発音チャンネルを解放したりする処理である。なお、ここで実行される処理の一部であるキーオン時のアサイン処理は、押鍵キーを発音チャンネルへ割り当てる処理のみであり、実際に押鍵キーに基づくピッチデータをアサインされた発音チャンネルに設定し発音指示を行う動作は、後述する発音処理（S909）において実行される。

【0044】次に、機能キー701（図7参照）であるデモ演奏キーが押されたときは、外部メモリ部116からデモ演奏データ（シーケンサデータ）が順次読み出され、S905と同様のキーアサイン処理などが行われる（S906）。また、リズムスタートキーが押されたときは、リズムデータが外部メモリ116から順次読み出され、S905と同様のキーアサイン処理などが行われる（S907）。

【0045】その後、以下に述べるタイマー処理が行われる（S908）。すなわち、後述するインタラプトタイマー処理（S1012）でインクリメントされている時間データの時間値が判別され、デモ演奏制御用に順次読み出される時間制御用のシーケンサデータ又はリズム演奏制御用に読み出される時間制御用のリズムデータと比較されることにより、S906のデモ演奏又はS907のリズム演奏を行う場合の時間制御が行われる。

【0046】更に、発音処理S909では、上記各ステッ

ブS905～S907でアサインされ発音開始されるべき発音チャンネルにピッチデータを設定し、また、発音中の発音チャンネルのピッチデータを予め設定されたエンベロープに従って時間的に変化させ、発音中の楽音のピッチに変化を付加するというピッチエンベロープ処理等が行われる。

【0047】更に、フロー1周準備処理が実行される(S910)。ここでは、上述のステップS909で新たにピッチデータが設定された発音チャンネルの状態を発音中に変えたり、逆にステップS905～S907で解放された発音チャンネルの状態を消音中に変える等の処理が行われる。

【0048】次に、図10のMCPUインタラプト処理につき説明する。MCPU101のインタラプト制御部203によりMCPU101に割り込みが掛かると、図9のメインフローチャートの処理が中断され、図10のMCPUインタラプト処理の実行が開始される。この場合、MCPUインタラプト処理のプログラムにおいて、図9のメインフローのプログラムで書き込みが行われるレジスタ等については、内容の書き換えが行われないように制御される。これにより、通常のインタラプト処理の開始時と終了時に行われるレジスタの退避と復帰の処理は不要となり、図9のメインフローチャートの処理とMCPUインタラプト処理との間の移行が迅速に行われる。

【0049】図10においては、まず、MCPUインタラプト処理において音源処理が開始される(S1011)。この音源処理は後述する図13に示される。上述の動作と同時に、MCPU101のインタラプト制御部203からSCPU102のRAMアドレス制御部305に対してSCPUリセット解除信号A(図1参照)が出力され、SCPU102において、図11のSCPUインタラプト処理の実行が開始される。

【0050】そして、MCPUインタラプト処理における音源処理(S1011)とほぼ同時に、SCPUインタラプト処理において音源処理が開始される(S1101)。このように、MCPU101とSCPU102の両者が並行して音源処理を実行することによって、1つのCPUで音源処理を実行する場合に較べて音源処理の処理速度が約2倍になる。

【0051】続いて、MCPU101では、S1012のインタラプトタイマー処理の後、SCPU102からSCPUインタラプト処理の終了信号が来るのを待っている(S1013)。なお、インタラプトタイマー処理では、図10のインタラプト処理が一定のサンプリング周期毎に実行されることを利用して、RAM206(図2)上の特には図示しない時間データの値がインクリメントされる。すなわち、この時間データの値を見れば時間経過がわかる。このようにして得られる時間データは、前述したように、図9のメインフローのタイマー処理S908における時間制御に用いられる。

【0052】図11のSCPUインタラプト処理におけるステップS1101の音源処理が終了すると、SCPU102のコマンド解析部307からMCPU101のメモリアドレス制御部205に、SCPU処理終了信号B(図1参照)が入力する。これにより、図10のMCPUインタラプト処理でのステップS1013の判定がYESになる。

【0053】この結果、図1のデータバスDinを介してSCPU102で作られた波形データがMCPU101のRAM206に読み込まれる(S1014)。この場合、波形データはSCPU102のRAM306上の所定のバッファ領域(図17で後述するバッファB)に格納されているため、MCPU101のコマンド解析部207は、SCPU内部アドレス指定バスMaを介して、RAMアドレス制御部304に対して上記バッファアドレスを指定することにより、波形データの読み込みを行う。

【0054】そして、S1014'において、上記バッファ領域の内容がLeft D/A変換器部107及びRight D/A変換器部108のラッチ401(図5参照)にラッチされる。

【0055】次に、図12は、前述の図9と図10のフローチャートの処理の関係を概念的に示した流れ図であり、MCPU101とSCPU102が、音源処理をそれぞれ分担して行う様子を示している。

【0056】まず、ある処理A(以下、処理B、C、・・・、Fも同じ)が実行される(S1201)。この処理は、図9のメインフローチャートの例えば「機能キー処理」、や「鍵盤キー処理」などに対応する。その後、MCPUインタラプト処理とSCPUインタラプト処理に入り、同時にMCPU101とSCPU102による音源処理が開始される(S1202、S1203)。そして、SCPU102でのSCPUインタラプト処理の終了時に、SCPU処理終了信号BがMCPU101に入力する。MCPUインタラプト処理では、SCPUインタラプト処理より早く音源処理が終了し、SCPUインタラプト処理の終了を待っている。そして、MCPUインタラプト処理においてSCPU処理終了信号Bが識別されると、SCPU102で生成された波形データがMCPU101に送られてMCPU101で生成された波形データとまとめられ、Left D/A変換器部107及びRight D/A変換器部108に出力される。その後、メインフローチャートの処理Aの次の処理Bに戻る。

【0057】以上のような動作が、全ての発音チャンネル(MCPU101とSCPU102で実行中の発音チャンネル)に対する音源処理が行われながら繰り返される(S1204～S1216)。そして、この繰り返し処理は、楽音の発音中続けられる。

【0058】図13は、MCPUインタラプト処理のステップS1011又はSCPUインタラプト処理のステップS1101で実行される音源処理の動作フローチャートである。

まず、RAM206又はRAM306の波形データ加算

用の領域がクリアされる (S1316)。次に、発音チャンネルの1チャンネル毎に音源処理が行われ (S1317~S1324)、最後に8チャンネル目の音源処理が終了した時点で、RAM206上の所定のバッファ領域B (後述する図17) に8チャンネル分が加算された波形データが得られる。

<音源処理におけるデータ構成>次に、図10のS1011及び図11のS1101で実行される音源処理の具体例について説明する。

【0059】本実施例では、MCPU 101とSCPU 102の両CPUが、最大8チャンネルづつの音源処理を分担することは前述した。この最大8チャンネル分の音源処理用のデータは、図14に示すように、MCPU 101、SCPU 102のRAM206、306内の発音チャンネル別の領域に設定される。

【0060】また、同RAMには、図17に示すように、全発音チャンネル共通のバッファBが確保されている。この場合、図14の各発音チャンネル領域には、後に詳述するような操作によって、図15に概念的に示すように、それぞれの音源方式が設定でき、その音源方式が設定されたら、図15及び図16に示すような各音源方式のデータフォーマットで、図14の各発音チャンネルの各領域にデータが設定される。なお、本実施例では、後述するように、各発音チャンネルに異なる音源方式を割り当てることが可能である。

【0061】図15及び図16の各音源方式のデータフォーマットを示すテーブル1において、Aは、音源処理時に波形データが読み出される場合に指定されるアドレスを表し、AI、A1及びA2が現在アドレスの整数部で、外部メモリ116 (図1) の波形データが格納されているアドレスに直接対応する。また、AFは現在アドレスの小数部であり、外部メモリ116から読み出された波形データの補間に用いられる。

【0062】次のAEはエンドアドレス、ALはループアドレスをそれぞれ表す。また、つぎのP1、P1及びP2はピッチデータの整数部、PFはピッチデータの小数部を表す。例を示すと、P1=1、PF=0は原音のピッチを、P1=2、PF=0は1オクターブ上のピッチを、また、P1=0、PF=0.5は、1オクターブ下のピッチをそれぞれ表す。

【0063】次のXPは前回のサンプルデータを、XNは次回のサンプルデータの格納を表す (後述する)。また、Dは隣接する2つのサンプルデータ間の大きさの差分値を表し、Eはエンベロープ値である。更に、Oは出力値である。

【0064】その他の種々の制御データについては、後述の各音源方式の説明の際に説明する。以上説明したような図15、図16に示すようなデータがMCPU 101、SCPU 102のそれぞれのRAM206、306に確保され、後述する音源方式が決まると、図14に示す

各チャンネル毎に図15、図16のフォーマットで、データが設定される。

【0065】以下、このようなデータ構成を用いて実行される各音源方式の音源処理について順次説明する。ここで、音色等の設定に応じて、各発音チャンネル毎にPCM、DPCM、FM又はTM等の異なる音源方式を割り当てることができる。そして、音色等の設定が行われた場合、8チャンネル分の音源方式が定まった音源処理プログラムが一括して、外部メモリ116 (図1) からプログラムメモリ201 (図2) 又はプログラムRAM301にロードされる。これにより、各発音チャンネル毎に、自動的に音源方式が定まった音源処理プログラムが実行されることになる。以下、これらの音源処理プログラムとして実行される各音源方式の原理について説明する。なお、音源処理プログラムの転送動作については後述する。

<PCM方式による音源処理>図18は、PCM方式による音源処理が実行される場合の動作フローチャートである。フロー中の各変数は、MCPU 101又はSCPU 102のRAM206、306上の図14のいずれかの発音チャンネル領域に記憶される図15、図16のテーブル1のPCMフォーマットの各データである。

【0066】外部メモリ116 (図1) 上のPCM波形データが記憶されているアドレスのうち、現在の処理の対象とされる波形データが記憶されているアドレスを図21に示す(AI, AF)とする。

【0067】まず、現在のアドレスにピッチデータ(P1, PF)が加算される(S1901)。このピッチデータは、図7の鍵盤キー702等において押鍵操作された鍵の種類に対応している。

【0068】そして、加算されたアドレスの整数部AIが変わったか否かが判定される(S1902)。判定がNOならば、図21のアドレス(AI+1)及びAIにおけるそれぞれのサンプルデータXNとXPとの差である差分値Dを用いて、 $D \times AF$ なる演算処理により、アドレスの小数部AFに対応する補間データ値Oが計算される(S1907)。なお、差分値Dは、今回以前のインタラプトタイミングにおける音源処理により求まっている (後述するS1906参照)。

【0069】そして、上記補間データ値Oにアドレスの整数部AIに対応するサンプルデータXPが加算され、現在のアドレス(AI, AF)に対応する新しいサンプルデータO (図21のXQに相当する) が得られる(S1908)。

【0070】この後、このサンプルデータにエンベロープ値Eが乗算され(S1909)、得られたOの内容がMCPU 101又はSCPU 102のRAM206又は306内の波形データバッファB (図17参照) に加算される(S1910)。

【0071】その後、図9のメインフローに戻り、次の

サンプリング周期でインタラプトが掛かって、図18の音源処理の動作フローチャートがふたたび実行され、現在アドレス(AI, AF)にピッチデータ(PI, PF)が加算される(S1901)。

【0072】以上の動作がアドレスの整数部AIが変わる(S1902)まで繰り返される。この間、サンプルデータXP及び差分値Dは更新されず、補間データOのみがアドレスAFに応じて更新され、その都度サンプルデータXQが得られる。

【0073】次に、S1901で現在アドレス(AI, AF)にピッチデータ(PI, PF)が加算された結果、現在アドレスの整数部AIが変化したら(S1902)、アドレスAIがエンドアドレスAEに達しているか又は越えているか否かが判定される(S1903)。

【0074】判定がYESならば、次のループ処理が行われる。すなわち、エンドアドレスAEを越えた分のアドレス(AI - AE)がループアドレスALに加算され、それにより得られた新しい現在アドレスの整数部AIからループ再生が開始される(S1904)。エンドアドレスAEとは、PCM波形データの最後の波形サンプルデータが記憶されている外部メモリ116(図1)上のアドレスである。また、ループアドレスALとは、演奏者が波形の出力を繰り返したい位置のアドレスであり、上記動作により、PCM方式で周知のループ処理が実現される。

【0075】S1903の判定がNOならば、上記ステップS1904の処理は実行されない。次に、サンプルデータの更新が行われる。ここでは、外部メモリ116(図1参照)から、新しく更新された現在アドレスAIと1つ手前のアドレス(AI - 1)に対応する各サンプルデータが、それぞれXN及びXPとして読み出される(S1905)。

【0076】更に、今までの差分値が、更新した上記XNとXPとの差分値Dに更新される(S1906)。これ以後の動作は前述した通りである。

【0077】以上のようにして、1発音チャンネル分のPCM方式による波形データが生成される。

<DPCM方式による音源処理>次に、DPCM方式について説明する。

【0078】まず、図22を用いて、DPCM方式の動作原理の概略を説明する。同図において、外部メモリ116(図1)のアドレスAIに対応するサンプルデータXPは、アドレスAIの1つ前の特には図示しないアドレス(AI - 1)に対応するサンプルデータとの差分値から求めた値である。

【0079】外部メモリ116(図1)のアドレスAIには、次の差分値Dが書き込まれているので、次のアドレスのサンプルデータは $XP + D$ で求められ、これが新たなサンプルデータXPとしておきかわる。この場合、現在アドレスを、図22のようにAFとすれば、現在アド

レスAFに対応するサンプルデータは、 $XP + D \times AF$ で求まる。

【0080】このように、DPCM方式では、現在のアドレスと、次のアドレスに対応するサンプルデータ間の差分値Dが外部メモリ116(図1)から読み出され、現在のサンプルデータに加算されて、次のサンプルデータが求められることにより、順次波形データが作成される。

【0081】このようなDPCM方式を採用すると、隣接する標本間の差分値が一般に小さい音声や楽音等のような波形を量子化する場合、通常のPCM方式に比較して、はるかに少ないビット数で量子化を行える。

【0082】以上のDPCM方式による音源処理が実行される場合の動作フローチャートを図19、図20に示す。フロー中の各変数は、MCPU101のRAM206又はSCPU102のRAM306上の図14のいずれかの発音チャンネル領域に記憶される図15のテーブル1のDPCMフォーマットの各データである。

【0083】外部メモリ116(図1)上のDPCM差分波形データが記憶されているアドレスのうち、現在の処理の対象とされるデータが記憶されているアドレスを、図22に示す(AI, AF)とする。

【0084】まず、現在アドレス(AI, AF)に、ピッチデータ(PI, PF)が加算される(S2001)。そして、加算されたアドレスの整数部AIが変化したか否かが判定される(S2002)。判定がNOならば、図22のアドレスAIにおける差分値Dを用いて、 $D \times AF$ なる演算処理により、アドレスの小数部AFに対応する補間データ値Oが演算される(S2014)。なお、差分値Dは、今回以前のインタラプトタイミングにおける音源処理により求まっている(後述するS2006とS2010参照)。

【0085】次に、上記補間データ値Oにアドレスの整数部AIに対応するサンプルデータXPが加算され、現在アドレス(AI, AF)に対応する新しいサンプルデータO(図22のXQに対応)が得られる(S2015)。

【0086】この後、このサンプルデータにエンベロープ値Eが乗算され(S2016)、得られたOの内容がMCPU101のRAM206又はSCPU102のRAM306内の波形データバッファB(図17参照)に加算される(S2017)。

【0087】その後、図9のメインフローに戻り、次のサンプリング周期でインタラプトが掛かり、図19、図20の音源処理の動作フローチャートが再び実行され、現在アドレス(AI, AF)にピッチデータ(PI, PF)が加算される(S2001)。

【0088】以上の動作が、アドレスの整数部AIに変化が生ずるまで繰り返される。この間、サンプルデータXP及び差分値Dは更新されず、補間データOのみがアドレスAFに応じて更新されて、その都度新たなサン

10

20

30

40

50

ルデータXQが得られる。

【0089】次に、S2001で現在アドレス(AI, AF)にピッチデータ(PI, PF)が加算された結果、現在アドレスの整数部AIが変化したら(S2002)、アドレスAIがエンドアドレスAEに達しているか或いは越えているかが判定される(S2003)。

【0090】判定がNOの場合、以下のS2004~S2007のループ処理により、現在アドレスの整数部AIに対応するサンプルデータが計算される。すなわち、まず、旧AIという変数(図15のテーブル1のDPCMの欄参照)には、現在アドレスの整数部AIが変化する前の値が格納されている。この値の格納は、後述するS2006又はS2013の処理の繰り返しにより実現される。この旧AIの値がS2006で順次インクリメントされながら、S2007で旧AIにより指示される外部メモリ116(図1)上の差分波形データがDとして読み出され、S2005において順次サンプルデータXPに累算される。そして、旧AIの値が変化後の現在アドレスの整数部AIに等しくなった時点で、サンプルデータXPの値は変化後の現在アドレスの整数部AIに対応する値となる。

【0091】このようにして、現在アドレスの整数部AIに対応するサンプルデータXPが求まると、S2004の判定がYESとなり、前述の補間値の演算処理(S2014)に移る。

【0092】上述の音源処理が各インタラプトタイミング毎に繰り返され、S2003の判定がYESに変化したら、次のループ処理に入る。まず、エンドアドレスAEを越えた分のアドレス(AI-AE)がループアドレスALに加算され、得られたアドレスが新たな現在アドレスの整数部AIとされる(S2008)。

【0093】以下、ループアドレスALからどれだけアドレスが進んだかによって、何回か差分値Dを累算する操作が繰り返されることにより、新たな現在アドレスの整数部AIに対応するサンプルデータXPが計算される。すなわち、まず、初期設定としてサンプルデータXPが予め設定されているループアドレスALにおけるサンプルデータXPL(図15のテーブル1のDPCMの欄参照)の値とされ、また、旧AIがループアドレスALの値とされる(S2009)。そして、以下のS2010~S2013の処理が繰り返される。すなわち、旧AIの値がS2013で順次インクリメントされながら、S2010で旧AIにより指示される外部メモリ116(図1)上の差分波形データがDとして読み出され、S2012において順次サンプルデータXPに累算される。そして、旧AIの値が新たな現在アドレスの整数部AIに等しくなった時点で、サンプルデータXPの値はループ処理後の新たな現在アドレスの整数部AIに対応する値となる。

【0094】このようにして、新たな現在アドレスの整数部AIに対応するサンプルデータXPが求まると、S2011の判定がYESとなって、前述の補間値の演算処理

(S2014)に移る。

【0095】以上のようにして、1発音チャンネル分のDPCM方式による波形データが生成される。

<FM変調方式による音源処理>次に、FM変調方式による音源処理について説明する。

【0096】FM変調方式では、通常、図24のOP1, OP2で示されるような、オペレータと呼ばれる同一機能を有するハードウェア又はソフトウェアのモジュールが用いられ、それらが同図に示されるような接続規則で相互に接続されることにより、楽音の生成が行われる。本実施例では、ソフトウェア処理によりFM変調方式を実現するものである。

【0097】図23は、2オペレータのFM変調方式による音源処理が実行される場合の動作フローチャートである。処理のアルゴリズムは図24で示される。また、フロー中の各変数は、MCPU 101又はSCPU 102のRAM 206、306上の図14のいずれかの発音チャンネル領域に記憶される、図16のテーブル1のFMフォーマットの各データである。

【0098】最初に、モジュレータであるオペレータ2(OP2)の処理が行われる。ピッチ処理については、PCM方式のように補間が行われないので、整数アドレスA2のみである。すなわち、外部メモリ116(図1)には変調用の波形データが十分に細かい歩進間隔で記憶されているものとする。

【0099】まず、現在アドレスA2にピッチデータP2が加算される(S2401)。次に、このアドレスA2にフィードバック出力F02が変調入力として加算され、新たなアドレスAM2が得られる(S2402)。フィードバック出力F02は、前回のインタラプトタイミングにおいて後述するS2405の処理が実行されることにより得られている。

【0100】更に、アドレスAM2(位相)に対応する正弦波の値が計算される。実際には、外部メモリ116

(図1)に正弦波データが記憶されており、上記アドレスAM2でその正弦波データをテーブル引きすることにより得られる(S2403)。

【0101】続いて、上記正弦波データにエンベロープ値E2が乗算され出力O2が得られる(S2404)。この後、この出力O2にフィードバック・レベルFL2が乗算されフィードバック出力F02が得られる(S2405)。この出力F02は、本実施例の場合、次のインタラプトタイミングにおけるオペレータ2(OP2)への入力とされる。

【0102】また、O2にモジュレーション・レベルML2が乗算されて、モジュレーション出力M02が得られる(S2406)。このモジュレーション出力M02は、オペレータ1(OP1)への変調入力になる。

【0103】次に、オペレータ1(OP1)の処理に移る。この処理は、フィードバック出力による変調入力がない

他は、殆ど上述のオペレータ2の場合と同じである。まず、オペレータ1の現在アドレスA1にピッチデータP1が加算され(S2407)、この値に上述のモジュレーション出力M02が加算されて新たなアドレスAM1が得られる(S2408)。

【0104】次に、このアドレスAM1(位相)に対応する正弦波の値が外部メモリ116(図1)から読み出され(S2409)、これにエンベロープ値E1が乗算され楽音波形出力O1が得られる(S2410)。

【0105】そして、これがRAM206(図2)又は306(図3)内のバッファB(図17参照)に加算されて(S2411)、1発音チャンネル分のFM変調処理を終了する。

<TM変調方式による音源処理>次に、TM変調方式に*

$$f_T(\omega) = 2/\pi \cdot \omega$$

$$f_T(\omega) = -1 + 2/\pi (3\pi/2 - \omega)$$

$$f_T(\omega) = -1 + 2/\pi (\omega - 3\pi/2)$$

また、fcは変形サイン波と呼ばれ、各位相角の領域毎に、異なるサイン波形データの記憶されている外部メモリ116(図1)を、搬送位相角 ωct でアクセスして※20

$$fc(t) = (\pi/2)\sin\omega ct$$

$$fc(t) = \pi - (\pi/2)\sin\omega ct$$

$$fc(t) = 2\pi + (\pi/2)\sin\omega ct$$

TM変調方式では、上述の如き関数 $fc(t)$ で生成される搬送信号に、変調信号 $\sin\omega mt$ を変調指数 $I(t)$ で示される割合で加算して得た加算信号により、前述の三角波関数が変調される。これにより、変調指数 $I(t)$ の値が0であれば正弦波を生成することができ、 $I(t)$ の値を大きくしてゆけば非常に深く変調された波形を生成することができる。ここで、変調信号 $\sin\omega mt$ の代わりに様々な信号を用いることができ、以下に述べるように、前回演算時の自分のオペレータ出力を一定のフィードバックレベルでフィードバックさせたり、他のオペレータの出力を入力させたりすることができる。

【0110】このような原理のもとでTM変調方式による音源処理が実行される場合の動作フローチャートを図25に示す。この場合も、図23のFM変調方式の場合と同様、2オペレータで音源処理を行う場合の例であり、処理のアルゴリズムは図26で示される。また、フロー中の各変数は、MCPU101又はSCPU102のRAM206、306上の図14のいずれかの発音チャンネル領域に記憶される、図16のテーブル1のTMフォーマットの各データである。

【0111】最初に、モジュレータであるオペレータ2(OP2)の処理が行われる。ピッチ処理については、PCM方式のように補間が行われないので、整数アドレスA2のみである。

【0112】まず、現在アドレスA2にピッチデータP2が加算される(S2601)。次に、変形サイン変換 fc により、上記アドレスA2(位相)に対応する変形サイン

*による音源処理について説明する。

【0106】まず、TM変調方式の原理について説明する。前述のFM変調方式は、

$$e = A \cdot \sin\{\omega ct + I(t) \cdot \sin\omega mt\}$$

なる演算式を基本とする。ここで、 ωct は搬送波位相角(搬送信号)、 $\sin\omega mt$ は変調波位相角(変調信号)、及び $I(t)$ は変調指数である。

【0107】これに対し、本実施例でTM変調方式と呼ぶ位相変調方式は、

$$e = A \cdot f_T\{fc(t) + I(t) \cdot \sin\omega mt\}$$

なる演算式を基本とする。ここで、 $f_T(t)$ は三角波関数であり、各位相角の領域毎に次のような関数で定義される(ただし、 ω は入力)。

【0108】

... (領域: $0 \leq \omega \leq \pi/2$)

... (領域: $\pi/2 \leq \omega \leq 3\pi/2$)

... (領域: $3\pi/2 \leq \omega \leq 2\pi$)

※得られる搬送信号生成関数である。各位相角の領域毎の fc は、次のように定義される。

【0109】

... (領域: $0 \leq \omega ct \leq \pi/2$)

... (領域: $\pi \leq \omega ct \leq 3\pi/2$)

... (領域: $3\pi/2 \leq \omega ct \leq 2\pi$)

波が外部メモリ116(図1)から読み出され、搬送信号がO2として生成される(S2602)。

【0113】続いて、搬送信号である上述のO2に、変調信号としてフィードバック出力F02(S2606)が加算され、新たなアドレスが得られてO2とされる(S2603)。フィードバック出力F02は、前回のインタラプトタイミングにおいて後述するS2606の処理が実行されることにより得られている。

【0114】そして、上述の加算アドレスO2に対応する三角波の値が計算される。実際には、外部メモリ116(図1)に前述した三角波データが記憶されており、上記アドレスO2でその三角波データをテーブル引きすることにより得られる(S2604)。

【0115】続いて、上記三角波データにエンベロープ値E2が乗算され出力O2が得られる(S2605)。この後、この出力O2にフィードバック・レベルFL2が乗算されフィードバック出力F02が得られる(S2607)。この出力F02は、本実施例の場合、次のインタラプトタイミングにおけるオペレータ2(OP2)への入力とされる。

【0116】また、O2にモジュレーション・レベルML2が乗算されて、モジュレーション出力M02が得られる(S2607)。このモジュレーション出力M02は、オペレータ1(OP1)への変調入力になる。

【0117】次に、オペレータ1(OP1)の処理に移る。この処理は、フィードバック出力による変調入力が無い他は、殆ど上述のオペレータ2の場合と同じである。ま

ず、オペレータ1の現在アドレスA1にピッチデータP1が加算され(S2608)、得られた値に対して前述の変形サイン変換が行われて搬送信号がO1として得られる(S2609)。

【0118】次に、このO1に上述のモジュレーション出力M02が加算されて新たなO1とされ(S2610)、この値O1が三角波変換され(S2611)、更にエンベロープ値E1が乗算されて楽音波形出力O1が得られる(S2612)。

【0119】これがRAM206(図2)又は306(図3)内のバッファB(図17参照)に加算され(S2613)、1発音チャンネル分のTM変調処理を終了する。以上、PCM、DPCM、FM、TMという4つの各種音源方式が実行される場合の動作原理を説明した。この中でFMとTMの2方式は変調方式であり、上述の例ではいずれも、図24、図26に示すアルゴリズムに基づく2オペレータによる処理であるが、実際に演奏時の音源処理は、オペレータの数がもっと多く、アルゴリズムはより複雑であってもよい。

＜音源処理プログラムの転送動作＞最後に、本発明に関連する音源処理プログラムの転送動作につき説明する。

【0120】前述したように、音色等の設定に応じて、各発音チャンネル毎にPCM、DPCM、FM又はTM等の異なる音源方式を割り当てることができる。そして、音色等の設定が行われた場合、8チャンネル分の音源方式が定まった音源処理プログラムが一括して、外部メモリ116(図1)からプログラムメモリ201(図2)又はプログラムRAM301にロードされる。これにより、各発音チャンネル毎にどの音源方式で楽音を生成するかという判別処理を行うことが不要となる。

【0121】この場合、MCPU101のプログラムメモリ201は、ROM部分とRAM部分とからなっており、ROM部分には予め図9のメインフローのプログラムが格納されている。これに対して、RAM部分には、音色設定時等において、図10のインタラプト処理のプログラムが図1の外部メモリ116から一括して転送される。また、SCPU102のプログラムRAM301にも、同様に図11のインタラプト処理のプログラムが転送される。

【0122】図27に外部メモリ116上に予め格納される各種音源処理プログラムのデータ配置を、図28にMCPU101内のプログラムメモリ201のRAM部分、又はSCPU102内のプログラムRAM301のデータ配置をそれぞれ示す。

【0123】今、演奏者が図7の機能キー701により音色設定等を行うと、本実施例による楽音波形発生装置は、例えば、MCPU101及びSCPU102上の楽音生成を行うための8つの発音チャンネルのそれぞれで、例えば全ての発音チャンネルでPCM方式により楽音生成を行う、又はTM方式により楽音生成を行う、或いは1～3

チャンネルでFM方式により、4～6チャンネルでTM方式により、7～8チャンネルでPCM方式により楽音生成を行う、というように機能することができる。

【0124】そして、外部メモリ116上には、例えば全ての発音チャンネルでPCM方式により楽音生成を行うための図13の動作フローチャートに基づく8チャンネル分連続の音源処理プログラム、又はTM方式により楽音生成を行うための8チャンネル分連続の音源処理プログラム、或いは1～3チャンネルでFM方式により、4～6チャンネルでTM方式により、7～8チャンネルでPCM方式により楽音生成を行うための8チャンネル分連続の音源処理プログラムが、図27のように、外部メモリ116上の独立したアドレス領域に格納されている。

【0125】この場合のデータ配置は、図27のように16ビット単位となっている。これは、外部メモリ116等として汎用のROMなどを使用した場合に、16ビット単位のものが多いので、それに対応可能とするためである。そして、16ビット単位で割り振られた各アドレス領域に、上述した音源処理プログラムが格納されている。

【0126】これに対して、プログラムメモリ201のRAM部分及びプログラムRAM301のデータ配置は、図28のように28ビット単位のネクストアドレス方式のプログラムデータとなっている。

【0127】そして、プログラムメモリ201上では、28ビット単位で割り振られた例えばアドレス1BFF～1FFF(16進表現)に、前述した8チャンネル分連続の音源処理プログラムが、外部メモリ116から転送される。なお、アドレス1BFFより前のアドレス部分は、図9のメインフローに対応するプログラムを格納するためのROM領域である。

【0128】ここで、外部メモリ116からプログラムメモリ201への音源処理のプログラムデータの転送時には、後述する図2のコマンド解析部207及びメモリアドレス制御部205が、両者のアドレスを同期させている。

【0129】以上の動作は、プログラムRAM301に対しても同様に行われる。但し、プログラムRAM301は音源処理を行うだけであるため、図9のメインフローを格納するためのROM領域はない。

【0130】次に、上記データ配置に基づく具体的な音源処理プログラムの転送動作について、図29の動作タイミングチャートを用いて説明する。この転送動作は、図2のMCPU101内のコマンド解析部207が中心となって、特に図示しない発振器から発生する図29(a)の原発振クロックから生成される同図(b),(c)の2相クロックCK2及びCK1に同期して実行するものとする。

【0131】まず、演奏者が図7の機能キー701で音色設定の動作を行うと、コマンド解析部207は、図9

のステップS903の機能キー処理において、図29(i)のタイミングで音源処理プログラム転送命令の実行を開始する。以下の動作は、全てコマンド解析部207が上記転送命令を実行する動作として実現される。

【0132】まず、コマンド解析部207は、設定された音色に対応する音源処理グループを判定し、それに対応する8チャンネル分の連続する音源処理プログラムが格納されている外部メモリ116上の先頭アドレスを決定する。

【0133】続いて、コマンド解析部207は、上記先頭アドレスから図29(d)のように順次インクリメントされるアドレス信号を、図2の外部メモリアクセス用アドレスバスインタフェース215、アドレスバスMA等を介して外部メモリ116に供給する。これによって、図27の16ビット単位で割り振られた、設定された音色に対応する8チャンネル分の連続する音源処理プログラムが格納されているアドレスが順次指定され、連続する16ビット及び12ビットの有効データからなるプログラムデータPRGが、図27のa, a', b, b', ...のように順次出力される。

【0134】これら16ビットの有効データ及び12ビットの有効データともに、図1の16ビットのデータバスMDから図2のMCPU 101の外部メモリデータバスインタフェース216に入力し、内部バスを介してプログラムメモリ201に転送される。

【0135】これと共に、コマンド解析部207は、図29(h)のようにハイレベルに立ち上がったアドレス切替信号Cを、図2のMCPU 101のメモリアドレス制御部205及び図3のSCPU 102のRAMアドレス制御部305とゲート回路314に出力する。メモリアドレス制御部205は、アドレス切替信号Cがハイレベルの場合には、楽音生成用のプログラムアドレスは生成せず、図29(e)のように順次インクリメントされるA0~A12の13ビットからなるアドレス信号ADRを出力する。このアドレス信号ADRは、プログラムアドレスデコーダ202に入力される。

【0136】また、アドレス切替信号Cがハイレベルであることにより、図3のゲート回路314は、外部メモリ116から図2のMCPU 101を介して転送されてきた音源処理のプログラムデータPRGを選択してプログラムRAM301に入力すると共に、上述のアドレス信号ADRを選択してRAMアドレス制御部305に送る。そして、同制御部305は、それをそのままRAMアドレスデコーダ302に入力する。これにより、図28の28ビット単位で割り振られたアドレス1BFF~1FFFが順次指定される。

【0137】同時に、コマンド解析部207は、図29(h)のようなアップダウン指示信号UDと、それに同期した図29(i)のようなライト信号~Wを、プログラムメモリ201及びゲート回路314からプログラムRAM

M301に供給する。

【0138】以上の制御動作によって、外部メモリ116から出力される音源処理のプログラムデータPRGの連続する16ビット及び12ビットの2データずつが、プログラムメモリ201及びプログラムRAM301上の各アドレス(13ビットのアドレス信号ADRで定まる)の下位16ビットと上位12ビット(アップダウン指示信号UDで定まる)に順次格納されてゆく。

【0139】以上のようにして、メモリアドレス制御部205がプログラムアドレスデコーダ202及びRAMアドレスデコーダ302に対して指定する13ビットのアドレス信号ADRが、図29(e)のように1FFFまでインクリメントされ、8チャンネル分の連続する音源処理のプログラムデータPRGの転送が終了したら、その旨が同制御部205からコマンド解析部207に通知される。これにより、同解析部207は、図29(d)のように外部メモリ116に対するアドレス信号の発生を終了し、同時に図29(f),(g),(h)のようにアップダウン指示信号UD、ライト信号~W及びアドレス切替信号Cをローレベルに戻す。

【0140】以上の処理動作が、コマンド解析部207が、図9のステップS903の機能キー処理において図29(i)の音源処理プログラム転送命令を実行することにより実現される。なお、上述の処理動作中は、図2のコマンド解析部207は、インタフェース制御部203に対して、音源処理のためのインタラプト信号を発生しないように制御を行う。

【0141】上述の動作以後、転送された音源処理プログラムに基づいて各発音チャンネル毎に任意の音源方式で楽音生成が実行される。上述の動作において、選択される8チャンネル分連続の音源処理プログラムによっては、プログラム容量が異なるが、本実施例では、プログラムメモリ201又はプログラムRAM301上には最大プログラム容量分の領域が確保され、上述のプログラム転送も無条件に最大プログラム容量分だけ実行される。従って、選択されたプログラムによっては、余分なプログラム部分が転送されることになるが、8チャンネル分連続の音源処理プログラムの最後には終了命令が格納されており、それが実行されることにより、以後の余分なプログラム部分は実行されないため問題は生じない。

【0142】また、上述の音源処理プログラムには、図10に対応するMCPU インタラプト処理のプログラム又は図11に対応するSCPU インタラプト処理のプログラムの部分まで含まれる。従って、プログラムメモリ201へのプログラム転送とプログラムRAM301へのプログラム転送とでは、厳密には前者の方が、図10のステップS1012~S1014'に対応する処理の分だけプログラムステップ数が多いが、上述の転送動作では、プログラムメモリ201及びプログラムRAM301の両者とも図10に対応する同一のプログラムが転送される。そ

ここで、同プログラムにおいて、ステップS1012にさしかかる境界部分に特殊命令を挿入しておき、図3のSCPU 102のコマンド解析部307は、プログラムRAM301から上記特殊命令を読み出したら、その時点でSCPU インタラプト処理を終了してMCPU 101に対してSCPU 処理終了信号B（図1参照）を出力する。従って、SCPU 102側では、上述の余分なプログラム部分は実行されない。

【0143】以上のようにして、音色設定時等に8チャンネル分の連続する音源処理プログラムを一括して外部メモリ116からプログラムメモリ201及びプログラムRAM301にロードすることにより、各発音チャンネル毎にどの音源方式で楽音を生成するかという判別処理を行うことが不要となり、図10又は図11のインタラプト処理の実行時間を短縮化できる。これにより、演奏情報を処理するための図9のメインフローのプログラムの実行時間を多くとることができ、楽音生成の応答性能を向上させることが可能となる。

＜他の実施例の態様＞上述の実施例では、演奏者が図7の機能キー701で音色設定の動作を行ったときに、図9のステップS903の機能キー処理において音源処理プログラム転送命令を実行するようにしたが、音色設定以外の操作を行った場合に実行するようにしてもよい。また、電源オン時等に自動的に実行される構成にすることもできる。

【0144】また、以上説明してきた本発明の一連の実施例では、図1に示される如く、MCPU 101とSCPU 102という2つのCPUが異なる発音チャンネルを分担して処理するようにしたが、CPUの数は1つでも、また、3つ以上でもよい。

【0145】更に、図2のMCPU 101の入力ポート210には、図7及び図8のような楽器操作部のほかに様々な操作部を接続することが可能であり、これにより種々の形態の電子楽器を実現できる。また、他の電子楽器からの演奏情報を入力して音源処理のみを行う音源モジュールとして実現することも容易である。

【0146】

【発明の効果】本発明によれば、プログラム実行手段は、所定時間間隔毎に各発音チャンネルの楽音生成を行う場合、音色設定時等に音源処理プログラム転送制御手段により予め実行用プログラム記憶手段に転送された全発音チャンネル分連続した音源処理プログラムを一括して実行することが可能となる。

【0147】これにより、各発音チャンネル毎にどの音源方式で楽音を生成するかという判別処理を行うことが不要となり、分岐命令の数を削減することが可能となるため、同時に発音可能な発音チャンネル数を増加させることができる等、楽器の性能を向上させることが可能となる。

【図面の簡単な説明】

【図1】本発明による実施例の全体構成図である。

【図2】マスターCPUの内部構成図である。

【図3】スレーブCPUの内部構成図である。

【図4】従来のD/A変換器部の構成図である。

【図5】本実施例によるD/A変換器部の構成図である。

【図6】D/A変換におけるタイミングチャートである。

【図7】機能キーと鍵盤キーの配置図である。

【図8】鍵盤キーの説明図である。

【図9】メイン動作フローチャートである。

【図10】MCPU インタラプト処理の動作フローチャートである。

【図11】SCPU インタラプト処理の動作フローチャートである。

【図12】メイン動作フローチャートとインタラプト処理の関係を示す概念図である。

【図13】音源処理の動作フローチャートである。

【図14】RAM上の発音チャンネル別の記憶領域を示す図である。

【図15】RAM上の音源方式別のデータフォーマットの構成図（その1）である。

【図16】RAM上の音源方式別のデータフォーマットの構成図（その2）である。

【図17】RAM上のバッファ領域を示す図である。

【図18】PCM方式による音源処理の動作フローチャートである。

【図19】DPCM方式による音源処理の動作フローチャート（その1）である。

【図20】DPCM方式による音源処理の動作フローチャート（その2）である。

【図21】PCM方式で差分値Dと現在アドレスAFを用いて補間値XQを求める場合の原理説明図である。

【図22】DPCM方式で差分値Dと現在アドレスAFを用いて補間値XQを求める場合の原理説明図である。

【図23】FM方式による音源処理の動作フローチャートである。

【図24】FM方式による音源処理のアルゴリズムを示す図である。

【図25】TM方式による音源処理の動作フローチャートである。

【図26】TM方式による音源処理のアルゴリズムを示す図である。

【図27】外部メモリ116のデータ配置図である。

【図28】プログラムメモリ201、プログラムRAM301のデータ配置図である。

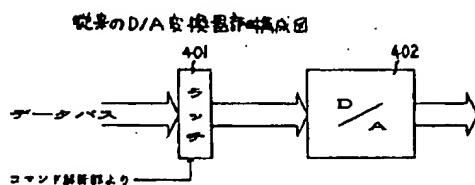
【図29】音源処理プログラム転送の動作タイミングチャートである。

【符号の説明】

50 101 マスターCPU

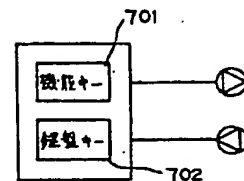
25	
102	スレーブCPU
103	MCPU 外部メモリアクセス用アドレスラッ
チ部	
104	SCPU 外部メモリアクセス用アドレスラッ
チ部	
105	アクセス用アドレス競合回避回路
106	外部メモリセクタ部
107	Left D/A変換器部
108	Right D/A変換器部
109	入力ポート
110	出力ポート
111、112	出力端子
113	レフト出力端子
114	ライト出力端子
115	外部メモリデータイン端子
116	外部メモリ
201	プログラムメモリ
202	メモリアドレスデコーダ
203	インタラプト制御部
204、205、304、305	RAMアドレス 20
制御部	
206、306	RAM
207、307	コマンド解析部
208、308	ALU部
209、309	乗算器
210	入力ポート
211	出力ポート
212	SCPU 内部RAMアドレス指定バス・イン
タフェース	
213	SCPU への書き込みデータバス・インタフ 30
ェース	
214	SCPU からの読み込みデータバス・インタ
フェース	
215	外部メモリアクセス用アドレスバス・イン
タフェース	

【図4】



26	
216	外部メモリデータバス・インタフェース
217	D/Aデータ転送バス・インタフェース
218	プログラムROM
219	イニシャル転送制御部
301	プログラムRAM
302	RAMアドレス制御部
303	MCPU によるSCPU 内部RAMアドレス指
定バス・インタフェース	
310	MCPU よりの書き込みデータバス・インタ
10	フェース
311	MCPU への読み出しデータバス・インタフ
ェース	
312	外部メモリアクセス用アドレスバス・イン
タフェース	
313	外部メモリデータバス・インタフェース
314	ゲート回路
401、501	ラッチ
402	D/A変換器
701	機能キー
702	鍵盤キー
A	SCPU リセット解除信号 (処理開始信号)
B	SCPU 処理終了信号
C	アドレス切替信号
Ma	SCPU 内部RAMアドレス指定バス
Dout	MCPU がSCPU へ書き込むデータバス
Din	MCPU がSCPU へ読み込むデータバス
MA	MCPU が外部メモリを指定するアドレスバス
SA	SCPU が外部メモリを指定するアドレスバス
MD	MCPU が外部メモリから読み込むデータバス
SD	SCPU が外部メモリから読み込むデータバス
PRG	プログラムデータ
ADR	アドレス信号
UD	アップダウン指示信号
~W	ライト信号

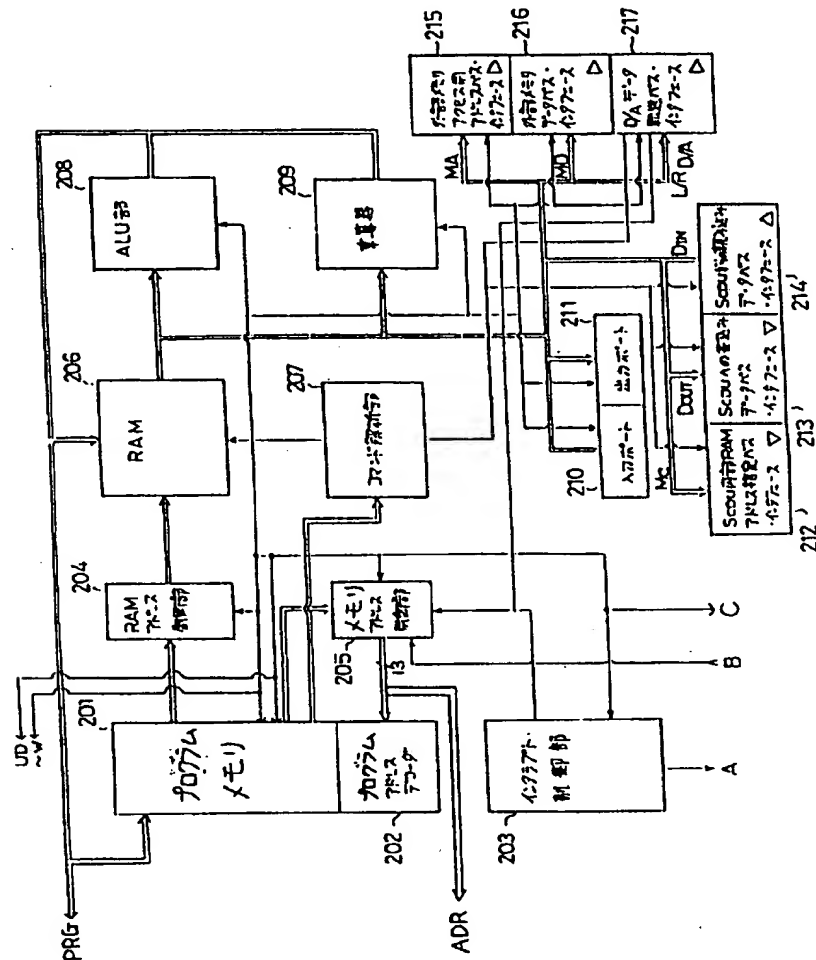
【図7】



機能キーと鍵盤キーの配置図

【图2】

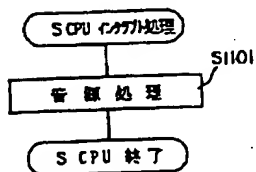
マスタ-CPUの内部構成図



【図 1 1】

【図 17】

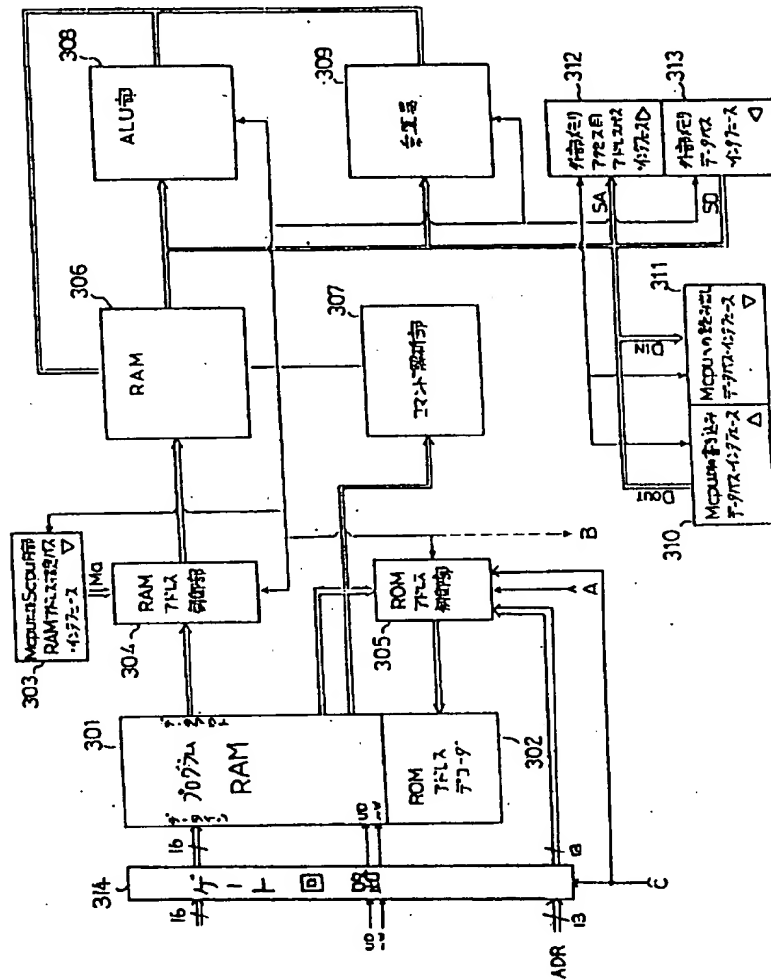
SCPU インタラプト処理の動作フローチャート



B (共通)

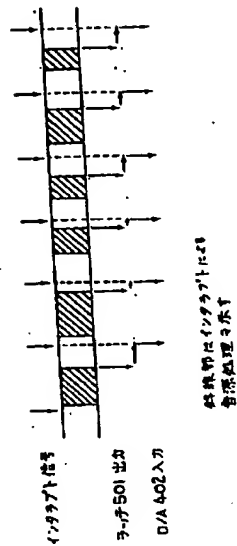
RAM上のバッファ領域を示す図

スレーブCPUの内部構成図



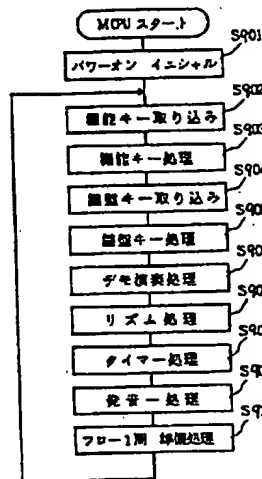
【図6】

D/A変換におけるタイミングチャート



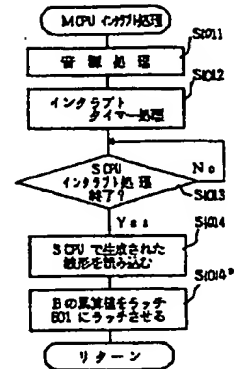
【図9】

メイン動作フローチャート



【図10】

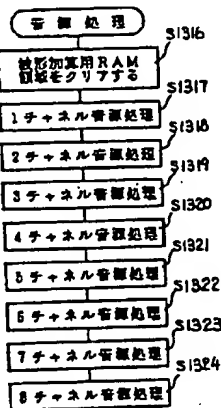
MOPU インタラプト処理の動作フローチャート



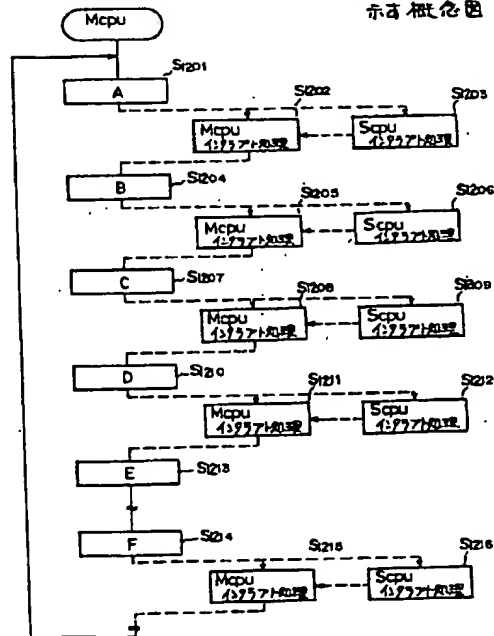
【図12】

【図13】

音源処理の動作フローチャート

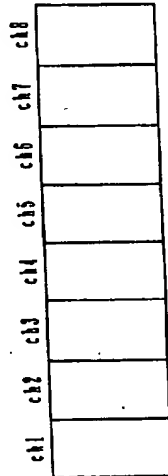


メイン動作フローチャートとインタラプト処理との関係を示す概念図



【図14】

RAM上の音声チャネル別の記憶領域を示す図



【図15】

RAM上の音源方式別のデータフォーマットの構成図
(その1)

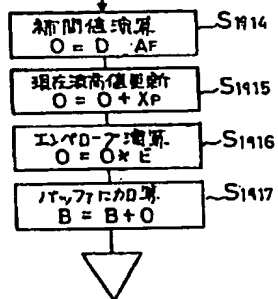
テーブル1

P C M	D P C M
A _i 現在アドレス整数部	A _i 現在アドレス整数部
A _f 現在アドレス小数部	A _f 現在アドレス小数部
A _e エンドアドレス整数部	A _e エンドアドレス整数部
A _l ループアドレス整数部	A _l ループアドレス整数部
P _f ビッチデータ小数部	P _f ビッチデータ小数部
P _i ビッチデータ整数部	P _i ビッチデータ整数部
X _p 前サンプルデータ	X _p 前サンプルデータ
X _n 後サンプルデータ	X _n 変化前の現在アドレス整数部
D サンプルデータ間差分値	D サンプルデータ間差分値
E エンベロープ値	E エンベロープ値
	X _n , A _l のサンプルデータ
0 出力	0 出力

【図20】

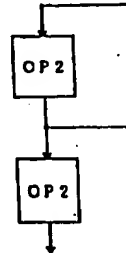
DPCM方式による音源処理の動作フローチャート(その2)

(図19から)



【図24】

FM方式による音源処理のアルゴリズムを示す図



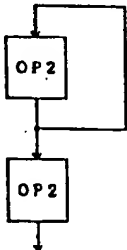
【図16】

RAM上の音源方式別のデータフォーマットの構成図
(その2)

F M	T M
A ₁ 現在アドレス (OP1)	A ₁
A ₂ 変位アドレス (OP2)	A ₂
<hr/>	
P ₁ ビッチデータ (OP1)	P ₁
P ₂ ビッチデータ (OP2)	P ₂
<hr/>	
E ₁ エンベロープ (OP1)	E ₁
E ₂ エンベロープ (OP2)	E ₂
M _{L1} モデル出力 (OP1)	M _{L1}
M _{L2} モデル出力 (OP2)	M _{L2}
F _{L1} フィルタ出力 (OP1)	F _{L1}
F _{L2} フィルタ出力 (OP2)	F _{L2}
O ₁ OP1出力	O ₁
O ₂ OP2出力	O ₂

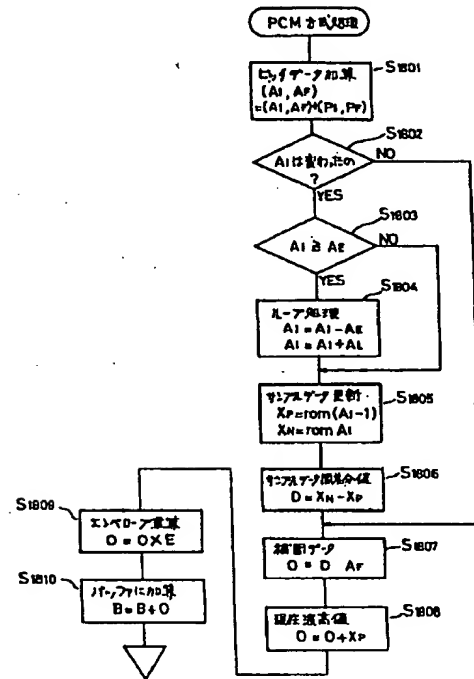
【図26】

TM方式による音源処理のアルゴリズムを示す図



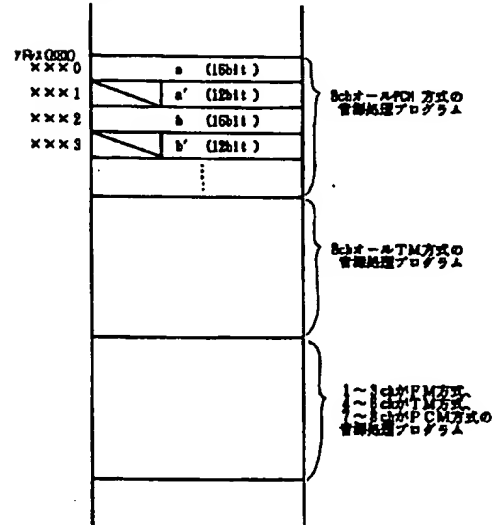
【図18】

PCM方式による音源処理の動作フローチャート



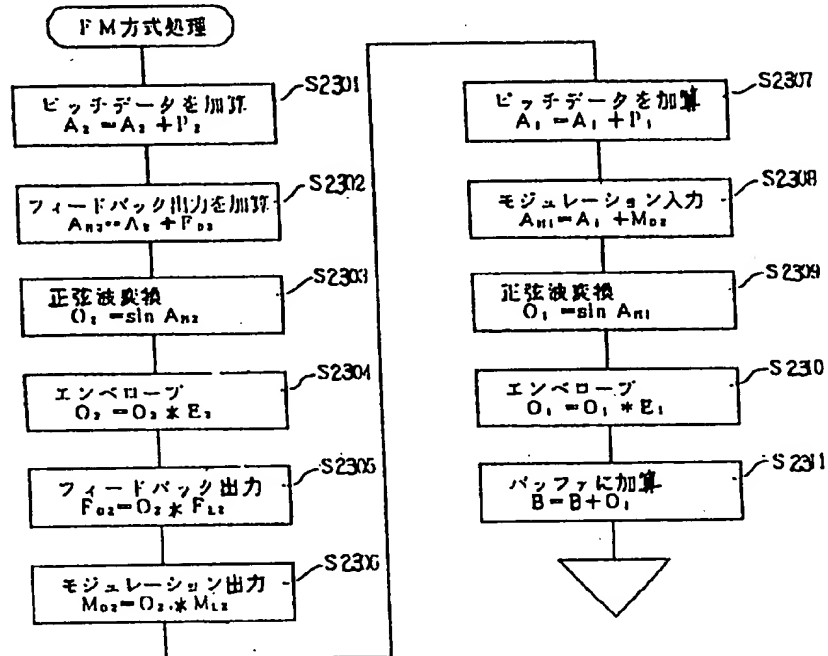
【図27】

外部メモリ116のデータ配列図

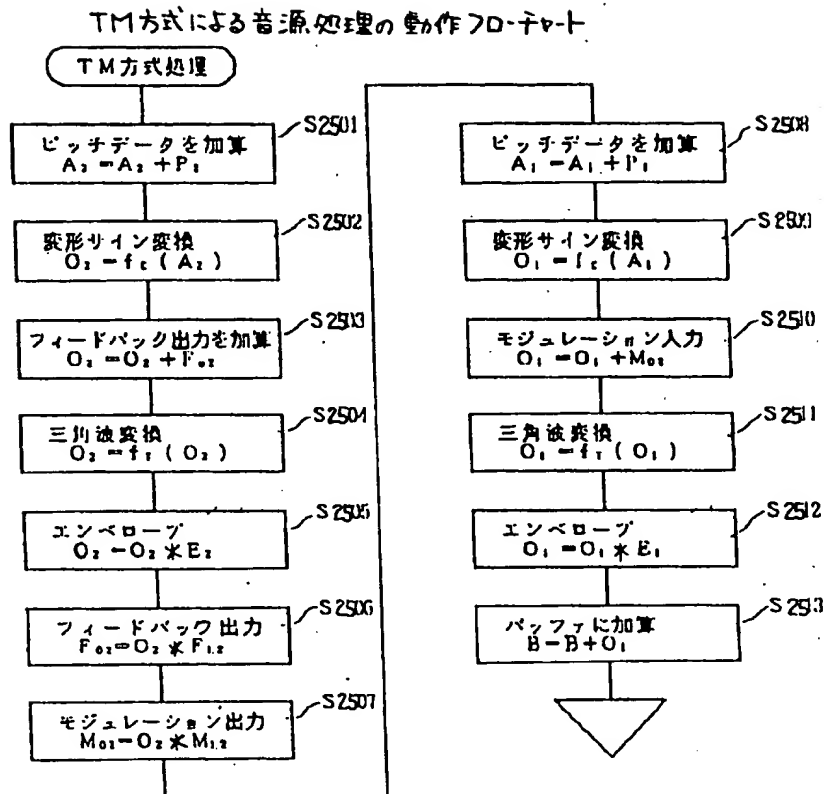


【図23】

FM方式による音源処理の動作フローチャート

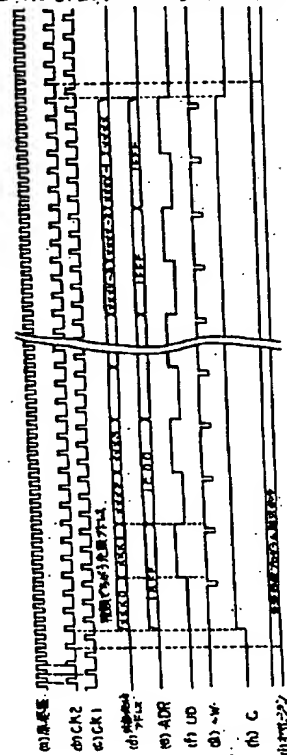


【図25】



【図29】

図29は、本発明の動作タイミング図を示す。



フロントページの続き

(72)発明者 斯波 康祐
東京都西多摩郡羽村町栄町3丁目2番1号
カシオ計算機株式会社羽村技術センター
内

(72)発明者 小倉 和夫
東京都西多摩郡羽村町栄町3丁目2番1号
カシオ計算機株式会社羽村技術センター
内

(72)発明者 細田 潤
東京都西多摩郡羽村町栄町3丁目2番1号
カシオ計算機株式会社羽村技術センター
内